IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPEAL BRIEF – 37 C.F.R § 1.192

U.S. Patent Application 10/697,052 entitled

"METHOD FOR DISCOVERING UNDECLARED AND FUZZY RULES IN DATABASES"

Real Party in Interest: International Business Machines Corporation

Related Appeals and Interferences:

None

Status of Claims:

Claims 1-10, 12-13, 16-17, and 22-24 are pending.

Claims 1-10, 12-13, 16-17, and 22-24 are rejected under 35 U.S.C. 101 for non-statutory subject matter.

Status of Amendments:

In response to non-final Office action mailed June 7, 2006, Applicants' Amendment was filed June 29, 2006. No amendments were filed in response to the final office action of 09/20/2006.

Summary of Claimed Subject Matter:

(NOTE: All citations are made with respect to the corresponding pre-grant publication, including the figures.)

The present invention, as per independent **claim 1**, provides for a computer-based method to perform query optimization by automatically finding and exploiting hidden, fuzzy algebraic constraints in a database (sec at least **figure 2**), said method comprising the steps of: (a) constructing one or more candidates of form $C=(a_1, a_2, P, \oplus)$, wherein a_1 and a_2 are numerical attributes associated with column values of data in said database, P is a pairing rule, and $\oplus$ is any of the following algebraic operators: +, -, ×, or / (see at least figure 2 and paragraph 76); (b)

constructing, for each candidate identified in (a), an algebraic constraint AC=(a_1, a_2, P, $\oplus$, I_1,…, I_k) by applying a segmentation technique, where I_1, …, I_k is a set of disjoint intervals and $k \geq 1$ (see at least **paragraph 76**), said step of constructing algebraic constraint further comprising the steps of: constructing a sample set W_C of an induced set Ω_C, wherein P is a join predicate between tables R and S and $\Omega_C = \{r.a_1 \oplus r.a_2 : r \in R\}$ when the pairing rule P is a trivial rule $\emptyset_R$ and $\Omega_C = \{r.a_1 \oplus s.a_2 : r \in R,\ s \in S, and\ (r,s)\ satisfies\ P\}$ (see at least **paragraphs 81** and **82**); sorting n data points in said sampled set W_C in increasing order as $x_1 \leq x_2 \leq \ldots \leq x_n$ and constructing a set of disjoint intervals I_1, …, I_k such that data in sample W_C falls within one of said disjoint intervals, wherein segmentation for constructing said set of disjoint intervals is specified via a vector of indices ($i(1)$, $i(2)$, …, $i(k)$) and the j^{th} interval is given by $I_j=[x_{i(j-1)+1}, x_{i(j)}]$ and length of Ij, denoted by Lj, is given by $L_j = x_{i(j)} - x_{i(j-1)+1}$ (see at least **paragraph 110**); and wherein the function for optimizing cost associated with said segmentation is $c(S) = wk + (1-w)\left[\frac{1}{\Delta}\sum_{j=1}^{k} L_j\right]$ (see at least **paragraph 112**), with w being a fixed weight between 0 and 1 and a segmentation that minimizes c is defined by placing adjacent points x_l and x_{l+1} in the same segment if and only if $x_{l+1}-x_l < d^*$, where $d^* = \Delta(w/(1-w))$ (see at least **paragraph 115**), and wherein said constructed algebraic constraints are used in query optimization.

The present invention according to dependent **claim 2**, in addition to the features of claim 1, teaches one or more pruning rules that are used to limit said number of constructed candidates (see at least **paragraphs 80**, **84**, **85**, and **97**).

The present invention according to dependent **claim 3**, in addition to the features of claim 2, teaches the pairing rule P representing either a trivial pairing rule $\o_R$ or a join between tables R and S and the pruning rules comprising any of, or a combination of the following (see at least paragraphs **97**, **98**, **99**, **100**, and **101**): pairing rule P is of form $R.a = S.b$ or of the form $\o_R$, and the number of rows in either table R or table S lies below a specified threshold value; pairing rule P is of form $R.a = S.b$ with $a \in K$ and the number of distinct values in $S.b$ divided by the number of values in $R.a$ lies below a specified threshold value, wherein K is a set comprising key-like columns among all columns in said database; pairing rule P is of form $R.a = S.b$, and one or both of R and S fails to have an index on any of its columns; or pairing rule P is of form $R.a = S.b$ with $a \in K$, and $S.b$ is a system-generated key.

The present invention according to dependent **claim 4**, in addition to the features of claim 1, further comprises the steps of: identifying a set of useful algebraic constraints via one or more pruning rules; and partitioning data into compliant data and exception data (see at least **paragraphs 65** and **146-151**).

The present invention according to dependent **claim 5**, in addition to the features of claim 4, comprises the steps of: receiving a query; modifying said query to incorporate identified constraints; and combining results of modified query executed on data in said database and said original query executed on exception data (see at least **paragraphs 79** and **80**).

The present invention according to dependent **claim 6**, in addition to the features of claim 4, comprises partitioning that is done by incrementally maintained materialized views, partial indices, or physical partitioning of the table (see at least **paragraph 30**).

The present invention according to dependent **claim 7**, in addition to the features of claim 2, comprises the pruning rules comprising any of, or a combination of the following: a_1 and a_2 are not comparable data types; the fraction of NULL values in either a_1 or a_2 exceeds a specified threshold; or either column a_1 or a_2 is not indexed (see at least **paragraphs 106**, **107**, and **108**).

The present invention according to dependent **claim 8**, in addition to the features of claim 1, comprises the step of constructing one or more candidates further comprises the steps of: generating a set P of pairing rules; and for each pairing rule $P \in \mathrm{P}$, systematically considering possible attribute pairs (a_1, a_2) and operators $\oplus$ with which to construct candidates (see at least **paragraph 85**).

The present invention according to dependent **claim 9**, in addition to the features of claim 8, further comprises the steps of: initializing P to be an empty set; adding a trivial pairing rule of the form $\emptyset_R$ to said set P for each table R in said database; and generating and adding nontrivial pairing rules to said set P based upon identifying matching columns via an inclusion dependency, wherein a column b is considered a match for column a if: data in columns a and b are of a comparable type; or either (i) column a is a declared primary key and column b is a declared foreign key for the primary key, or (ii) every data value in a sample from column b has a matching value in column a (see at least **paragraphs 86**, **87**, **91**, **92**, and **96**).

The present invention according to dependent **claim 10**, in addition to the features of claim 4, comprises the steps of: initializing P to be an empty set; adding a trivial pairing rule of the form $\emptyset_R$ to said set P for each table R in said database; and generating a set K of key-like columns from among all columns in said database with each column in set K belonging to a predefined set of types T, said set K comprising declared primary key columns, declared unique key columns, and undeclared key columns, wherein said primary keys or declared unique keys are compound keys of form $a = (a_1, \ldots, a_m) \in T^m$ for $m>1$; adding nontrivial pairing rules to said set P based upon identifying matching compound columns via an inclusion dependency wherein, given a compound key $(a_1, \ldots, a_m) \in K$, a compound column b is considered a component wise match for compound column a if: data in compound columns a and b are of a comparable type; or either (i) compound column a is a declared primary key and compound column b is a declared foreign key for the primary key, or (ii) every data value in a sample from compound column b has a matching value in compound column a (see at least **paragraphs 86, 87, 91, 92, 93, 94**, and **96**).

The present invention according to dependent **claim 12**, in addition to the features of claim 1, teaches expanding widths associated with said intervals to avoid additional sampling required to increase right end point to equal maximum value in Ω_C (see at least **paragraph 111**).

The present invention according to dependent **claim 13**, in addition to the features of claim 1, teaches the approximation of the size of said sampled set via the following iterative steps: (a) given a k-segmentation, setting counters i=1 and k=1; (b) selecting a sample size

$n=n^*$, wherein $n^*(k) \approx \frac{\chi^2_{1-p}(2-f)}{4f} + \frac{k}{2}$, wherein p is the probability that at least a fraction of points in Ω_C that lie outside the intervals is at most f; (c) obtaining a sample based on (b), computing algebraic constraints, and identifying a number k' of bump intervals; and (d) if $n \geq n^*(k')$ or $i = i_{max}$, then utilizing sample size in (b); else setting counters $k=k'$ and $i=i+1$, and returning to step (b) (see at least **paragraphs 128**, **131**, **132**, **133**, and **134**).

The present invention according to dependent **claim 16**, in addition to the features of claim 1, teaches the method being implemented across networks (see at least **paragraph 157**).

The present invention according to dependent **claim 17**, in addition to the features of claim 16, teaches said across networks element comprising any of, or a combination of the following: local area network (LAN), wide area network (WAN), or the Internet (see at least **paragraph 157**).

The present invention according to independent **claim 22**, teaches an article of manufacture comprising a computer usable medium having computer readable program code embodied therein which implements a method to perform query optimization by automatically finding and exploiting hidden, fuzzy algebraic constraints in a database (see at least **figure 2** and **paragraph 157**), said method comprising the steps of: (a) computer readable program code constructing one or more candidates of form $C=(a_1, a_2, P, \oplus)$, wherein a_1 and a_2 are numerical attributes associated with column values of data in said database, P is a pairing rule, and $\oplus$ is any of the following algebraic operators: +, -, ×, or / (see at least **paragraph 76**); (b) computer readable program code constructing, for each candidate identified in (a), an algebraic constraint

AC=(a_1, a_2, P, $\oplus$, I_1,..., I_k) (see at least **paragraph 69**) by applying a segmentation technique, where I_1, ..., I_k is a set of disjoint intervals and $k \geq 1$, said step of constructing algebraic constraint further comprising the steps of: constructing a sample set W_C of an induced set Ω_C, wherein P is a join predicate between tables R and S and $\Omega_C = \{r.a_1 \oplus r.a_2 : r \in R\}$ when the pairing rule P is a trivial rule $\emptyset_R$ and $\Omega_C = \{r.a_1 \oplus s.a_2 : r \in R, s \in S, and\ (r,s)\ satisfies\ P\}$ (see at least **paragraph 81**); sorting n data points in said sampled set W_C in increasing order as $x_1 \leq x_2 \leq \ldots \leq x_n$ and constructing a set of disjoint intervals I_1, ..., I_k such that data in sample W_C falls within one of said disjoint intervals, wherein segmentation for constructing said set of disjoint intervals is specified via a vector of indices ($i(1)$, $i(2)$, ..., $i(k)$) and the j^{th} interval is given by $I_j=[x_{i(j-1)+1}, x_{i(j)}]$ and length of Ij, denoted by Lj, is given by $L_j = x_{i(j)} - x_{i(j-1)+1}$ (see at least **paragraph 110**); and wherein the function for optimizing cost associated with said segmentation is $c(S) = wk + (1-w)\left[\frac{1}{\Delta}\sum_{j=1}^{k} L_j\right]$ (see at least **paragraph 112**) with w being a fixed weight between 0 and 1 and a segmentation that minimizes c is defined by placing adjacent points x_l and x_{l+1} in the same segment if and only if $x_{l+1}-x_l < d^*$, where $d^* = \Delta(w/(1-w))$ (see at least **paragraph 115**), and wherein said constructed algebraic constraints are used in query optimization (see at least **abstract**).

The present invention according to dependent **claim 23**, in addition to the features of claim 22, further comprises: computer readable program code identifying a set of useful algebraic constraints via heuristics comprising a set of pruning rules; and computer readable

program code partitioning data into compliant data and exception data (see at least **paragraphs 65** and **146-151**).

The present invention according to dependent **claim 24**, in addition to the features of claim 23, further comprises: computer readable program code aiding in receiving a query; computer readable program code modifying said query to incorporate identified constraints; and computer readable program code combining results of modified query executed on data in said database and said original query executed on exception data (see at least **paragraphs 79** and **80**).

Grounds of Rejection to be Reviewed on Appeal:

1. Claims 1-10, 12-13, 16-17, and 22-24 are rejected under 35 U.S.C. §101 for non-statutory subject matter. Was a proper rejection made under 35 U.S. C. §101 using existing USPTO guidelines?

ARGUMENT:

REJECTIONS UNDER 35 U.S.C. § 101

The U.S. Patent & Trademark Office's Examination Guidelines for Computer-Related Inventions provide that computer-related process claims, to be statutory, must either: (A) result in a physical transformation outside the computer for which a practical application in the technological arts is either disclosed in the specification or would have been known to a skilled artisan, or (B) be limited to a practical application within the technological arts that produces a useful, concrete, and tangible result.

Under the first safe harbor provision, a process is statutory if it requires physical acts to be performed outside the computer independent of and following the steps to be performed by a programmed computer, where those acts involve the manipulation of tangible physical objects and result in the object having a different physical attribute or structure.

Under the second safe harbor provision, if the claim produces a useful, concrete, and tangible result, the claim is limited to a practical application, and is therefore statutory. State Street Bank & Trust Co. v. Signature Financial Group Inc., 47 USPQ2d 1596, 1601-02 (Fed. Cir. 1998). Further, the focus is not on whether the steps taken to achieve a particular result are useful, tangible, and concrete, but rather on whether the final result achieved by the claimed invention is useful, tangible, and concrete.

With respect to pending claims 1-10, 16, 18, 20, 22-31, and 34-38, the examiner on page

2 of the office action of 09/20/2006 states that the preamble phrase "to perform query optimization by automatically finding and exploiting hidden, fuzzy algebraic constraints" is an "exercise at best". The Examiner on the same page of the office action questions "How does one exploit hidden fuzzy algebraic constraints and what are the benefits of doing so?" The Examiner appears to ignore the body of the claim while relying solely on the preamble to making such statements. For example, the body of the independent claims 1 and 22, for example, *specifically* recites, in detail: (1) how algebraic constraints are constructed; and (2) that the constructed algebraic constraints are used in query optimization in a database scenario.

Claims 1 and 22, for example, teach constructing one or more candidates of form $C=(a_1, a_2, P, \oplus)$, wherein a_1 and a_2 are numerical attributes associated with column values of data in said database, P is a pairing rule, and $\oplus$ is any of the following algebraic operators: +, -, ×, or /. Claims 1 and 22 also specifically teaches constructing, for each candidate identified, an algebraic constraint $AC=(a_1, a_2, P, \oplus, I_1, \ldots, I_k)$ by applying a segmentation technique, where $I_1, \ldots, I_k$ is a set of disjoint intervals and $k \geq 1$, wherein said constructed algebraic constraints are used in query optimization.

Hence, it is seen that claims 1 and 22 both address, with specificity, how algebraic constraints are constructed for use in query optimization. Hence, Applicants maintain that the Examiner's comment about "how does one exploit hidden fuzzy algebraic constraints" are without merit as the claim address with specificity how algebraic constraints are constructed for each constructed candidate of form $C=(a_1, a_2, P, \oplus)$ in a database. The Examiner's second

concern regarding what the benefits are also addressed in claims 1 and 22 as they specifically recite the use of such constructed algebraic constraints in "query optimization".

Further, the Examiner states on page 4 of the office action of 09/20/2006 that the final result of claims 1 and 22 are not "useful, tangible, and concrete". The Examiner goes on to question that "what purpose does finding and exploiting hidden, fuzzy, algebraic constraints in view of real world functionality". However, in making such statements, the Examiner appears to ignore specific statements both in the preamble and the body of claims 1 and 22 that the claims are directed to "query optimization". Further, the Examiner, on page 3 of the office action of 09/20/2006 answers the posed question by stating that "the invention is a process for optimizing a query". Given the Examiner's own statement, Applicants assert that claims 1 and 22 specifically outline a method for "query optimization" and therefore provides the Examiner's requested "real world functionality".

Additionally, Applicants wish to note that the U.S.P.T.O. has recognized "query optimization" as an utility/practical application with "real world functionality" by assigning it as specific class/subclass combination 707 (Data Processing: Database and File Management or Data Structures)/5 (..Query augmenting and refining). Applicants, therefore, assert that the final result of query optimization according to independent claims 1 and 22 is useful, tangible and concrete.

The above arguments for independent claims 1 and 22 substantially apply to the pending

dependent claims as they inherit all the features of the claim from which they depend. Thus, Applicants respectfully assert that pending claims 1-10, 16, 18, 20, 22-31, and 34-38 produces a useful, concrete, and tangible result, and that the claims are limited to a practical application, and are therefore statutory. Hence, Applicants respectfully contend that the Examiner erroneously issued an 35 U.S.C. 101 rejection with respect to the pending claims.

SUMMARY

As has been detailed above, pending claims 1-10, 16, 18, 20, 22-31, and 34-38 cover statutory subject matter. It is believed that this case is in condition for allowance and reconsideration thereof and early issuance is respectfully requested.

As this Appeal Brief has been timely filed within the set period of response, no petition for extension of time or associated fee is required. However, the Commissioner is hereby authorized to charge any deficiencies in the fees provided, to include an extension of time, to Deposit Account No. 09-0441.

Respectfully submitted by
Applicant's Representative,

/randywlacasse/

Randy W. Lacasse
Reg. No. 34368

1725 Duke Street, Suite 650
Alexandria, VA 22314
(703) 838-7683

Claims Appendix:

1. **(Original)** A computer-based method to perform query optimization by automatically finding and exploiting hidden, fuzzy algebraic constraints in a database, said method comprising the steps of:

(a) constructing one or more candidates of form $C=(a_1, a_2, P, \oplus)$, wherein a_1 and a_2 are numerical attributes associated with column values of data in said database, P is a pairing rule, and $\oplus$ is any of the following algebraic operators: +, -, ×, or /;

(b) constructing, for each candidate identified in (a), an algebraic constraint $AC=(a_1, a_2, P, \oplus, I_1,\ldots, I_k)$ by applying a segmentation technique, where $I_1, \ldots, I_k$ is a set of disjoint intervals and $k \geq 1$, said step of constructing algebraic constraint further comprising the steps of:

constructing a sample set W_C of an induced set Ω_C, wherein P is a join predicate between tables R and S and $\Omega_C = \{r.a_1 \oplus r.a_2 : r \in R\}$ when the pairing rule P is a trivial rule $\emptyset_R$ and $\Omega_C = \{r.a_1 \oplus s.a_2 : r \in R,\ s \in S, and\ (r,s)\ satisfies\ P\}$;

sorting n data points in said sampled set W_C in increasing order as $x_1 \leq x_2 \leq \ldots \leq x_n$ and constructing a set of disjoint intervals $I_1, \ldots, I_k$ such that data in sample W_C falls within one of said disjoint intervals, wherein segmentation for constructing said set of disjoint intervals is specified via a vector of indices $(i(1), i(2), \ldots, i(k))$ and the j^{th} interval is given by $I_j=[x_{i(j-1)+1}, x_{i(j)}]$ and length of Ij, denoted by Lj, is given

by $L_j = x_{i(j)} - x_{i(j-1)+1}$; and

wherein the function for optimizing cost associated with said segmentation is

$c(S) = wk + (1-w)\left[\frac{1}{\Delta}\sum_{j=1}^{k} L_j\right]$ with w being a fixed weight between 0 and 1 and a segmentation that minimizes c is defined by placing adjacent points x_l and x_{l+1} in the same segment if and only if $x_{l+1}-x_l < d^*$, where $d^* = \Delta(w/(1-w))$, and

wherein said constructed algebraic constraints are used in query optimization.

2. (Original) A compute-based method as per claim 1, wherein one or more pruning rules are used to limit said number of constructed candidates.

3. (Original) A computer-based method as per claim 2, wherein said pairing rule P represents either a trivial pairing rule $\emptyset_R$ or a join between tables R and S and said pruning rules comprise any of, or a combination of the following:

pairing rule P is of form $R.a = S.b$ or of the form $\emptyset_R$, and the number of rows in either table R or table S lies below a specified threshold value;

pairing rule P is of form $R.a = S.b$ with $a \in K$ and the number of distinct values in $S.b$ divided by the number of values in $R.a$ lies below a specified threshold value, wherein K is a set comprising key-like columns among all columns in said database;

pairing rule P is of form $R.a = S.b$, and one or both of R and S fails to have an

index on any of its columns; or

pairing rule P is of form $R.a = S.b$ with $a \in K$, and $S.b$ is a system-generated key.

4. **(Original)** A computer-based method as per claim 1, wherein said method further comprises the steps of:

identifying a set of useful algebraic constraints via one or more pruning rules; and

partitioning data into compliant data and exception data.

5. **(Original)** A computer-based method as per claim 4, wherein said method further comprises the steps of:

receiving a query;

modifying said query to incorporate identified constraints; and

combining results of modified query executed on data in said database and said original query executed on exception data.

6. **(Original)** A computer-based method as per claim 4, wherein said partitioning is done by incrementally maintained materialized views, partial indices, or physical partitioning of the table.

7. **(Original)** A computer-based method as per claim 2, wherein said pruning rules comprise any of, or a combination of the following:

a_1 and a_2 are not comparable data types;

the fraction of NULL values in either a_1 or a_2 exceeds a specified threshold; or

either column a_1 or a_2 is not indexed.

8. (Original) A computer-based method as per claim 1, wherein said step of constructing one or more candidates further comprises the steps of:

generating a set P of pairing rules; and

for each pairing rule $P \in$ P, systematically considering possible attribute pairs (a_1, a_2) and operators $\oplus$ with which to construct candidates.

9. (Original) A computer-based method as per claim 8, wherein said step of generating a set P of pairing rules further comprises the steps of:

initializing P to be an empty set;

adding a trivial pairing rule of the form $\emptyset_R$ to said set P for each table R in said database; and

generating and adding nontrivial pairing rules to said set P based upon identifying matching columns via an inclusion dependency, wherein a column b is considered a match for column a if:

data in columns a and b are of a comparable type; or

either (i) column a is a declared primary key and column b is a declared foreign key for the primary key, or (ii) every data value in a sample from column b has a matching value in column a.

10. (Original) A computer-based method as per claim 8, wherein said step of generating a set P of pairing rules further comprises the steps of:

initializing P to be an empty set;

adding a trivial pairing rule of the form $\emptyset_R$ to said set P for each table R in said database; and

generating a set K of key-like columns from among all columns in said database with each column in set K belonging to a predefined set of types T, said set K comprising declared primary key columns, declared unique key columns, and undeclared key columns, wherein said primary keys or declared unique keys are compound keys of form $a = (a_1, \ldots, a_m) \in T^m$ for $m>1$;

adding nontrivial pairing rules to said set P based upon identifying matching compound columns via an inclusion dependency wherein, given a compound key $(a_1, \ldots, a_m) \in K$, a compound column b is considered a component wise match for compound column a if:

data in compound columns a and b are of a comparable type; or

either (i) compound column a is a declared primary key and compound column b is a declared foreign key for the primary key, or (ii) every data value in a sample from compound column b has a matching value in compound column a.

11. (Cancelled)

12. (Previously Presented) A computer-based method as per claim 1, wherein widths associated with said intervals are expanded to avoid additional sampling required to increase right end point to equal maximum value in Ω_C.

13. (Previously Presented) A computer-based method as per claim 1, wherein size of said sampled set is approximated via the following iterative steps:

(a) given a k-segmentation, setting counters i=1 and k=1;

(b) selecting a sample size $n=n^*$, wherein $n^*(k) \approx \frac{\chi^2_{1-p}(2-f)}{4f} + \frac{k}{2}$, wherein p is the probability that at least a fraction of points in Ω_C that lie outside the intervals is at most f;

(c) obtaining a sample based on (b), computing algebraic constraints, and identifying a number k' of bump intervals; and

(d) if $n \geq n^*(k')$ or $i = i_{max}$, then utilizing sample size in (b); else setting counters $k=k'$ and $i=i+1$, and returning to step (b).

14 – 15 (Cancelled)

16. (Original) A computer-based method as per claim 1, wherein said method is implemented across networks.

17. (Original) A computer-based method as per claim 16, wherein said across networks element comprises any of, or a combination of the following: local area network (LAN), wide area network (WAN), or the Internet.

18 – 21 (Cancelled

22. (Previously Presented) An article of manufacture comprising a computer usable medium having computer readable program code embodied therein which implements a method to perform query optimization by automatically finding and exploiting hidden, fuzzy algebraic constraints in a database, said method comprising the steps of:

(a) computer readable program code constructing one or more candidates of form $C=(a_1, a_2, P, \oplus)$, wherein a_1 and a_2 are numerical attributes associated with column values of data in said database, P is a pairing rule, and $\oplus$ is any of the following algebraic operators: +, -, ×, or /;

(b) computer readable program code constructing, for each candidate identified in (a), an algebraic constraint AC=$(a_1, a_2, P, \oplus, I_1,\ldots, I_k)$ by applying a segmentation technique, where $I_1, \ldots, I_k$ is a set of disjoint intervals and $k \geq 1$, said step of constructing algebraic constraint further comprising the steps of:

constructing a sample set W_C of an induced set Ω_C, wherein P is a join predicate between tables R and S and $\Omega_C = \{r.a_1 \oplus r.a_2 : r \in R\}$ when the pairing rule P is a trivial

rule $\o_R$ and $\Omega_C = \{r.a_1 \oplus s.a_2 : r \in R, s \in S, and\ (r,s)\ satisfies\ P\}$;

sorting n data points in said sampled set W_C in increasing order as $x_1 \leq x_2 \leq \ldots \leq x_n$ and

constructing a set of disjoint intervals $I_1, \ldots, I_k$ such that data in sample W_C falls within one of said disjoint intervals, wherein segmentation for constructing said set of disjoint intervals is specified via a vector of indices $(i(1), i(2), \ldots, i(k))$ and the j^{th} interval is given by $I_j=[x_{i(j-1)+1}, x_{i(j)}]$ and length of Ij, denoted by Lj, is given by $L_j = x_{i(j)} - x_{i(j-1)+1}$; and

wherein the function for optimizing cost associated with said segmentation is $c(S) = wk + (1-w)\left[\frac{1}{\Delta}\sum_{j=1}^{k} L_j\right]$ with w being a fixed weight between 0 and 1 and a segmentation that minimizes c is defined by placing adjacent points x_l and x_{l+1} in the same segment if and only if $x_{l+1}-x_l < d^*$, where $d^* = \Delta(w/(1-w))$, and

wherein said constructed algebraic constraints are used in query optimization.

23. (Original) An article of manufacture as per claim 22, wherein said medium further comprises:

computer readable program code identifying a set of useful algebraic constraints via heuristics comprising a set of pruning rules; and

computer readable program code partitioning data into compliant data and

exception data.

24. (Original) An article of manufacture as per claim 23, wherein said medium further comprises:

computer readable program code aiding in receiving a query;

computer readable program code modifying said query to incorporate identified constraints; and

computer readable program code combining results of modified query executed on data in said database and said original query executed on exception data.

25 – 38 (Cancelled)

Evidence Appendix

None

Related Proceedings Appendix

None